

## CLAIMS

### What is claimed is:

- 1 1. A method for canceling a task in a computer system, the computer system comprising a  
2 task scheduler for managing a plurality of tasks using at least one task queue, wherein the  
3 task scheduler is arranged to free resources assigned to a cancelled task of the plurality of  
4 tasks when the cancelled task reaches the top of any of the at least one task queue, the  
5 method comprising steps of:  
6       - identifying a task from the plurality of tasks as a cancelled task;  
7       - actively prioritizing the identified task to the top of its corresponding task  
8       queue; and  
9       - allowing the task scheduler to free resources assigned to the identified task.
- 1 2. The method of claim 1, wherein the step of identifying comprises a step of calling a Cancel  
2 function associated with the identified task.
- 1 3. The method of claim 1, wherein the steps of identifying and prioritizing are performed by  
2 calling a CancelAndRemove function associated with the identified task.
- 1 4. The method of claim 1, wherein the step of prioritizing further comprises a step of setting a  
2 NextExecution parameter of the identified task to a value near zero.
- 1 5. The method of claim 4, wherein the step of prioritizing further comprises a step of  
2 updating an Index parameter associated with the identified task in accordance with the top  
3 of its corresponding task queue.
- 1 6. The method of claim 1, wherein the step of allowing further comprises notifying a memory  
2 garbage collector associated to the task scheduler.

- 1 7. A task scheduler within a computer system comprising:
  - 2 - at least one task queue capable of managing a plurality of tasks;
  - 3 - a prioritizing module capable of:
    - 4 - actively prioritizing a task within the plurality of tasks to the top of
    - 5 its corresponding task queue, wherein the task has been identified
    - 6 as a cancelled task; and
  - 7 - a memory garbage collector capable of:
    - 8 - freeing resources assigned to the identified task when the
    - 9 identified task reaches the top of any of the at least one task queue.
- 1 8. The task scheduler of claim 7, wherein the prioritizing module is further capable of setting  
2 a NextExecution parameter of the identified task to a value near zero.
- 1 9. The task scheduler of claim 7, wherein the prioritizing module is further capable of  
2 updating an Index parameter associated with the identified task in accordance with the top  
3 of its corresponding task queue.
- 1 10. The task scheduler of claim 7, wherein the prioritizing module is further capable of  
2 notifying the memory garbage collector.